IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Murali Basavaiah

Atty Docket No.: ANDIP037/425584

Application No.: 10/726,269

Examiner: Unelus, E.

Filed: December 1, 2003

Group: 2181

Title: APPARATUS AND METHOD FOR PERFORMING FAST FIBRE CHANNEL WRITE OPERATIONS OVER RELATIVELY HIGH LATENCY NETWORKS

Confirmation No: 3368

CERTIFICATE OF EFS-WEB TRANSMISSION
I hereby certify that this correspondence is being transmitted electronically through EFS-WEB to the Commissioner for Patents, P.O. Box 1450 Alexandria, VA 22313-1450 on **10/05/09**

Signed: /Jennifer K. Hardin-Cole/
Jennifer K. Hardin-Cole

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

The Applicants respectfully request review of the final rejection in the above-identified application. No amendments are being filed with this request. This request is being filed with a Notice of Appeal. The review is requested for the reasons stated below.

Claims 1-3, 5-20 and 24-31 are currently pending in the application. Claims 4 and 21-23 were previously cancelled. Claims 1-3, 5-20 and 24-31 were rejected by the present Office Action. The rejections are respectfully traversed.

REJECTIONS UNDER 35 U.S.C. § 103(a)

Claims 1-3, 5-20, and 24-31 were rejected under 35 U.S.C. 103(a) as being unpatentable over Mullendore et al. (U.S. Publication No. 2003/0185154) in view of Kaul et al. (U.S. Publication No. 2005/0050211). The Applicants respectfully submit that Mullendore and Kaul do not teach the elements of the claims, either alone or in combination.

The present claims recite certain mechanisms for improving data transmission across a network, which mechanisms operate in part by manipulating OX_ID and RX_ID values. For example, independent claims 24, 27, 29, and 30 variably recite

> receiving a write command at a switch, the write command . . . including an originator exchange identifier (OX_ID) with an assigned value and an uninitialized receiver exchange identifier (RX_ID) with a default value.

Independent claims 24, 27, 29, and 30 further variably recite

> initializing the receiver exchange identifier (RX_ID) by assigning a value to the RX_ID;
>
> sending a transfer ready command including the initialized RX_ID to the host prior to receiving a transfer ready command from the target, wherein sending the transfer ready command to the host allows the switch to operate as a proxy for the target;
>
> modifying the originator exchange identifier (OX_ID) of the write command to generate a modified write command; and
>
> forwarding the modified write command to the target.

Independent claim 1 contains similar (though slightly different) claim language relating to assigning values to OX_ID and RX_ID.

As is well understood in the art, OX_ID and RX_ID are fields of a frame header under, for example, the Fibre Channel (FC) protocol. Where they are used, as stated in the Specification, OX_ID and RX_ID are used by hosts and targets within a network to keep track of various transactions or "exchanges" between each other:

> [0015] To identify an FC device, Fibre Channel Identifiers (FCIDs) are used. A transaction between an FC host and a target is referred to as an exchange. In a typical Fibre Channel network, there are many Hosts and targets. Each Host may initiate many read and/or write operations. *For the hosts and targets within a network to keep track of the various transactions between each other,* two fields are available in the Fibre Channel header for all SCSI Command, Data, Response, and Transfer Ready frames. *The first field is called the Originator Exchange Identifier or OX_ID. The second field is called the Receiver Exchange Identifier or RX_ID. The Host relies on the OX_ID to maintain its local state and the target relies on the RX_ID to maintain its local state*." (Specif., para. [0015] (emphasis added).)

The Examiner admits that Mullendore, the primary reference, does not teach or suggest "a frame having a header with an OX_ID or RX_ID". (Office Action, pages 4 and 11.) However, the Examiner asserts that Kaul teaches these recitations. In relying upon Kaul, the Examiner refers only to the following two paragraphs of Kaul:

> [0023] Whenever a call terminal inside LAN 110 wants to send a packet outside LAN 110, it forwards the packet to NAT 108. The IP header of the packet uses the local address of the call terminal for the source address of the packet. NAT 108 receives the packet on its local interface, *modifies the IP header of the packet to change the source address to the global address of LAN* 110, and then sends the packet to network 112.
>
> [0024] Whenever a packet for a call terminal within LAN 110 is received by NAT 108 at its global address interface, it uses the combination of global address and the port number at which it received the data to map it to a local address and port number for the destination call terminal within LAN 110. Before forwarding the packet to the destination call terminal within LAN 110, *NAT 108 changes the destination address in the IP header from the global address to the local address of the destination call terminal in LAN 110*. Once this is done, NAT 108 forwards the packet to the appropriate destination call terminal in LAN 110. (Emphasis added).

However, these paragraphs do nothing more than describe fairly conventional processes for Network Address Translation (NAT). As shown in the above paragraphs, Kaul makes no mention of OX_ID or RX_ID, much less a frame header having such fields. While both Kaul's IP network address translation technology and the claimed invention replace or modify a value in a frame header prior to initiating communication with a different entity, the similarities end there. Kaul's network address translation pertains to IP addresses and is intended to address the problems posed by having a limited IP address space. It concerns translation of addresses at interfaces between a Local Area Network (LAN) and a wider network outside the LAN, an entirely different concern than the ones addressed by Mullendore and the present application. (*See* Kaul, Abstract and paras. [0023] and [0024] above.)

As discussed in the specification, a particular Host and target in a data storage system may have multiple transactions (for example, read and write requests) occurring between them at any given time. In various embodiments, the exchange identifiers are fields provided in the FC header to allow a Host and target to identify (i.e., "keep track of") different transactions between each other (typically read and write operations), and, in some embodiments, to provide a way of accessing and organizing such transactions or "exchanges". Accordingly, in various embodiments, the OX-ID and RX-ID identifiers serve a function quite different from that of destination and source addresses. They do not identify a particular device on a network as addresses do; they identify a particular transaction between two devices and the temporary roles of two nodes during such tranaction. They do not assist in the routing of data packets or frames to a destination; they assist Host and target devices in organizing and differentiating different transactions or "exchanges" within themselves.

Furthermore, the exchange identifiers are relatively short-lived, lasting only as long as a particular transaction, whereas the duration of a source or destination address is either static in

the case of Media Access Control (MAC) addresses or with statically assigned IP addresses, or dependent on factors completely independent from the transactions occurring between devices in the case of dynamically assigned IP addresses.

As a further consideration, please understand that certain independent claims require a switch to change an OX_ID field on one type of communication (a write command) and an RX_ID field on a different type of communication (a transfer ready command). Kaul makes no such distinction. Network address translation typically occurs at the gateway switch.

The Examiner contends that the global and the local IP addresses mentioned in Kaul "can be interpreted" as OX_ID and RX_ID identifiers. The Examiner states: "Similarly, Kaul discloses a data packet having a routing header identifying a source and destination target; in the same way that a RX_ID is used to specifies [sic] a target. *In other words, OX_ID and RX_ID are being interpreted as addresses for a source and a destination*." (Office Action, page 4 (emphasis added).) The Examiner states that "modifying the OX_ID" in the claims is "being equated to the NAT modifying a header to change the source address to the global address" and that "to initialize a receiver exchange identifier (RX_ID)" is "being equated to the NAT changing the global address to the local address". (Office Action, page 5.)

There are multiple problems with the Examiner's analogy, some of which were listed above. Additionally, the OX_ID and RX_ID cannot be analogized to source and destination addresses because, as recited in the independent claims, the frame headers at issue contain particular fields (i.e., host and target identifiers) separate from OX_ID and RX_ID, for identifying the source and destination of a communication. Claims 24, 27, 29, and 30 recite: "the write command specifying a *host identifier* corresponding to a host and a *target identifier* corresponding to a target, the write command also including an *originator exchange identifier (OX_ID)* with an assigned value and an uninitialized *receiver exchange identifier (RX_ID)* with a default value." (Emphases added). Claim 1 contains similar claim language.

The Examiner cites paragraph [0018] of Applicants' specification as showing that OX_ID and RX_ID can be analogized to source and destination addresses. That paragraph of the Specification states, among other things: "As previously noted, the OX_ID field 32 and the RX_ID field 34 are each 16 bits wide and *are used for identifying the originating Host and target device*." (Specif., para. [0018] (emphasis added)). If that statement were the only one in the Specification concerning the OX_ID and RX_ID fields, and the context was ignored, the Examiner might have an argument, though still on a very slender basis. However, in this case, that statement appears in the midst of four paragraphs (paragraphs [0015] through [0019]) discussing OX_ID and RX_ID and it is clear from the context that the Specification does not define OX_ID and RX_ID to merely identify the host and target. The Specification, as discussed at length above, makes clear that the OX_ID and RX_ID are used to identify particular

transactions or *exchanges* between a host and a target, and not the actual host and target outside the context of a transaction or exchange. (*See* Specif., para. [0015] – [0019].)

It is important to note that the pending claims achieve numerous advantages over the cited art, many of which are achieved in part through specific manipulations of the OX_ID and RX_ID fields. For example, as noted above, in various embodiments, switches use the OX_ID and/or RX_ID values to track specific exchanges between a host and target. For example, the Specification states: "[0010] Since the Switches send frames to the initiating Host independent of the target, the Switches manipulate the OX_ID and RX_ID fields in the Fibre Channel header of the various commands associated with the SCSI Write. The OX_ID and RX_ID fields are manipulated so as to trap the commands and so that the Switches can keep track of the various commands associated with the SCSI write." (Spec., para. [0010].) As would be known to those with skill in the art, modifications to the OX_ID and RX_ID fields do not usually occur at switches. (*See* Specif., para. [0016].) The OX_ID and RX_ID fields are typically used only by the host and the target. (Specif., [0016].) Thus, the techniques of the present invention intelligently operate to modify OX_ID or RX_ID values in order to achieve certain significant advantages and perform certain important functions. Furthermore, various embodiments contain an inventive feature in which switches intelligently operate to modify OX_ID or RX_ID values.

For at least the above reasons, the Applicants respectfully submit that Mullendore and Kaul do not teach the elements of the independent claims, either alone or in combination. In view of the foregoing, the Applicants respectfully request that the rejections of independent claims 1, 24, 27, 29, and 30, and their dependent claims, be withdrawn.

CONCLUSION

The Applicants believe that all pending claims are allowable. Should the Examiner believe that a telephone conference would expedite prosecution of this application, the Examiner is encouraged to contact the Applicants' representative at the telephone number set forth below.

Respectfully submitted,
Weaver Austin Villeneuve & Sampson LLP

/Jeffrey K. Weaver/

Jeffrey K. Weaver
Reg. No. 31,314

Weaver Austin Villeneuve & Sampson LLLP
P.O. Box 70250
Oakland, CA 94612-0250
(510) 663-1100

APPENDIX OF PENDING CLAIMS

1. (Previously Presented) An apparatus, comprising:

a port configured to receive a write command frame having a header with an OX_ID or RX_ID exchange identifier, as well as initiating Host and target identifiers;

a trapping mechanism configured to trap the write command frame; and

a processor configured to process the trapped write command frame by modifying the OX_ID of the write command frame header to include a new value of the OX_ID exchange identifier before sending the write command frame to the target;

wherein the processor is further configured to initialize a receiver exchange identifier (RX_ID) of a transfer ready command frame by assigning a value to the RX_ID and send the transfer ready command frame to the initiating Host before receiving a transfer ready command frame from the target.

2. (Previously Presented) The apparatus of claim 1, wherein the apparatus is an initiating Switch coupled to the Host in a first SAN.

3. (Previously Presented) The apparatus of claim 2, wherein the processor of the initiating Switch is further configured to modify the write command frame before forwarding the write command to the target.

4. (Cancelled)

5. (Previously Presented) The apparatus of claim 14, wherein the apparatus uses the value as a handle for accessing information pertaining to the write command session in a sessions ID table.

6. (Original) The apparatus of claim 2, wherein the processor of the initiating Switch is further configured to issue a Transfer Ready command to the Host.

7. (Previously Presented) The apparatus of claim 1, wherein the apparatus is further configured to use the value as the RX_ID for all communication related to the write frame between the apparatus and the Host.

8. (Previously Presented) The apparatus of claim 1, wherein the apparatus is further configured to use the value as the OX_ID in all communications between the apparatus and the target.

9. (Previously Presented) The apparatus of claim 2, wherein the initiating Switch is further configured to transfer additional data frames to the target when the initiating Switch receives a Transfer Ready command associated with the write command frame from the target.

10. (Previously Presented) The apparatus of claim 30, wherein the Switch is a target Switch coupled to the target.

11. (Previously Presented) The apparatus of claim 10, wherein the target Switch forwards the write command frame to the target.

12. (Previously Presented) The apparatus of claim 11, wherein the target Switch forwards data frames associated with the write command frame to the target after receiving a Transfer Ready command from the target.

13. (Original) The apparatus of claim 12, wherein the target Switch is further configured to buffer the data frames prior to receipt of the Transfer Ready command.

14. (Previously Presented) The apparatus of claim 12, wherein the target Switch is further configured to maintain a sessions ID table and to use the OX_ID of the write command frame as an index to the session corresponding to the write command.

15. (Previously Presented) The apparatus of claim 10, wherein the target Switch is further configured to modify the RX_ID value for all communication related to the write command frame between the target Switch and the Host.

16. (Original) The apparatus of claim 5, wherein the target Switch is further configured to modify the OX_ID value with communications between the target Switch and the target.

17. (Previously Presented) The apparatus of claim 1 wherein the apparatus is further configured to use the RX_ID value of trapped write commands to specify the amount of buffer

space needed for the write command and use the write command frame to request the needed buffer space.

18. (Previously Presented) The apparatus of claim 17, wherein the apparatus is further configured to use the RX_ID value of trapped write commands to specify the amount of buffer space larger than needed for the write command and use the additional buffer space for subsequent write commands so that the apparatus need not wait for a Transfer Ready command to transfer data related to the subsequent write command.

19. (Previously Presented) The apparatus of claim 1, wherein the apparatus is further configured to, in the event the apparatus does not have sufficient buffer space for the write command, to either:

(i) generate a busy status signal to the initiating Host;

(ii) place the write command on a pending wait list; or

(iii) forward the write command to the target.

20. (Previously Presented) The apparatus of claim 1, further comprising:

a first SAN including the apparatus;

a second SAN; and

an inter-SAN network connecting the first SAN and the second SAN.

21-23. (Canceled)

24. (Previously Presented) A method comprising:

receiving a write command at a switch, the write command specifying a host identifier corresponding to a host and a target identifier corresponding to a target, the write command also including an originator exchange identifier (OX_ID) with an assigned value and an uninitialized receiver exchange identifier (RX_ID) with a default value;

initializing the receiver exchange identifier (RX_ID) by assigning a value to the RX_ID;

sending a transfer ready command including the initialized RX_ID to the host prior to receiving a transfer ready command from the target, wherein sending the transfer ready command to the host allows the switch to operate as a proxy for the target;

modifying the originator exchange identifier (OX_ID) of the write command to generate a modified write command; and

forwarding the modified write command to the target.

25. (Previously Presented) The method of claim 24, further comprising configuring the switch to forward data frames associated with the write command received in response to the transfer Ready command to the target.

26. (Previously Presented) The method of claim 25, wherein a second switch between the switch and the target receives data frames associated with the write command and buffers the data frames until a transfer ready command is received from the target.

27. (Previously Presented) An apparatus comprising:

means for receiving a write command at a switch, the write command specifying a host identifier corresponding to a host and a target identifier corresponding to a target, the write command also including an originator exchange identifier (OX_ID) with an assigned value and an uninitialized receiver exchange identifier (RX_ID) with a default value;

means for initializing the receiver exchange identifier (RX_ID) to generate an initialized RX_ID by assigning a value to the_RX_ID;

means for sending a transfer ready command including the initialized RX_ID to the host prior to receiving a transfer ready command from the target, wherein sending the transfer ready command to the host allows the switch to operate as a proxy for the target;

means for modifying the originator exchange identifier (OX_ID) of the write command to generate a modified write command; and

means for forwarding the modified write command to the target.

28. (Previously Presented) The apparatus as recited in claim 1, wherein the apparatus is further configured to determine from the write command an amount of data to be written to the target, to ascertain whether it has sufficient storage space to buffer the amount of data, and to send the transfer ready command frame to the initiating Host before receiving the transfer ready command from the target if the apparatus has determined that it has sufficient storage space to buffer the amount of data.

29. (Previously Presented) A method comprising:

receiving a write command at a switch, the write command specifying a host identifier corresponding to a host and a target identifier corresponding to a target, the write command also including an originator exchange identifier (OX_ID) with an assigned value and an uninitialized receiver exchange identifier (RX_ID) with a default value;

forwarding the write command to the target;

initializing the receiver exchange identifier (RX_ID) to generate an initialized RX_ID by assigning a value to the RX_ID; and

sending a transfer ready command including the initialized RX_ID to the host prior to receiving a transfer ready command from the target, wherein sending the transfer ready command to the host allows the switch to operate as a proxy for the target.

30. (Previously Presented) An apparatus, comprising:

a processor; and

a memory, at least one of the processor or the memory being for:

receiving a write command at a switch, the write command specifying a host identifier corresponding to a host and a target identifier corresponding to a target, the write command also including an originator exchange identifier (OX_ID) with an assigned value and an uninitialized receiver exchange identifier (RX_ID) with a default value;

forwarding the write command to the target;

initializing the receiver exchange identifier (RX_ID) by assigning a value to the RX_ID and

sending a transfer ready command including the initialized RX_ID to the host prior to receiving a transfer ready command from the target, wherein sending the transfer ready command to the host allows the switch to operate as a proxy for the target.

31. (Previously Presented) The apparatus as recited in claim 1, wherein the trapping mechanism is configured to trap the write command frame if the write command frame designates a predetermined Host_ID and a predetermined target_ID.